REMARKS/ARGUMENTS

Claims 1-5, 7-10, and 12-14 are pending in the present application. With this amendment, claim 12 has been amended. Reconsideration of the claims is respectfully requested.

I. 35 U.S.C. § 112, First Paragraph

The Examiner has rejected claims 12-14 under 35 U.S.C. § 112, first paragraph, as failing to adequately comply with the enablement requirement. This rejection is respectfully traversed.

The Examiner states:

> Claim 12 recites the limitation "said buffering device" in claim 12, 11.7-8. There is insufficient antecedent basis for this limitation in the claim.
> As per claim 12, it appears unclear as to which "buffering device" the applicant is referring to, as there appears to have no prior recitation of "a buffering device" claim limitation.
> As per claims 13-14, dependent claims 13-14 are rejected at least due to dependency on the rejected independent claim 12.

Final Office Action dated February 7, 2007, pages 2-3.

Applicant traverses the rejection of claims 12-14 under 35 U.S.C. § 112, first paragraph, as failing to adequately comply with the enablement requirement.

Applicant has amended claim 12 to correct the antecedent basis.

This rejection has been overcome and should be withdrawn.

II. 35 U.S.C. § 103, Obviousness

II.A. Claims 1, 4-5, 12 and 14 over *O'Grady* in view of *Jaquette*

The Examiner has rejected claims 1, 4-5, 12 and 14 under 35 U.S.C. § 103(a) as being unpatentable over *O'Grady* et al., Method and Apparatus for Reordering Packet Data Units in Storage Queues for Reading and Writing Memory, U.S. Patent No. 6,757,791, dated June 29, 2004 (hereinafter "*O'Grady*") in view of *Jaquette* et al., ECC in Memory Arrays Having Subsequent Insertion of Content, U.S. Patent No. 6,009,547, dated December 28, 1999 (hereinafter "*Jaquette*"). This rejection is respectfully traversed.

The Examiner relies on *O'Grady* to teach the features of Applicant's claims but states that *O'Grady* does not teach "as said data is being written to each one of said plurality of blocks, calculating a running cyclical redundancy code (CRC) for each of said plurality of blocks; and when writing to each one of said plurality of blocks is completed, storing, for each one of said plurality of blocks, a final value of said running CRC that was calculated for each one of said plurality of blocks as a first CRC in a second memory on said buffering device".

On page 6 of the Final Office Action that was mailed February 7, 2007, the Examiner states that *Jaquette* teaches "calculating a first cyclical redundancy code (CRC) for each of said plurality of blocks (col. 4, ll. 24-27), wherein the first CRC corresponds to the ECC check symbols calculated; and storing the first CRC in a second memory on said buffering device (col. 5, ll. 13-26), wherein the first CRC corresponds to the ECC check symbols are stored in a different segment of the DRAM from where the data blocks are stored".

The combination of *O'Grady* and *Jaquette* does not render Applicant's claims obvious because the combination does not teach or suggest "as said data is being written to each one of said plurality of blocks, calculating a running cyclical redundancy code for each of said plurality of blocks; and when writing to each one of said plurality of blocks is completed, storing, for each one of said plurality of blocks, a final value of said running cyclical redundancy code that was calculated for each one of said plurality of blocks as a first cyclical redundancy code in a second memory on said buffering device".

Applicant claims calculating a running cyclical redundancy code for each block as data is being written to each block. In contradistinction, *Jaquette* waits until the data is formatted into blocks and then calculates a final ECC check symbol for the entire block of data. This ECC check symbol is not a running ECC check symbol that is calculated as data is being stored into a block.

> The blocks of the data file are supplied to memory manager 18 and byte level ECC check symbols are calculated by CRC generator 20 for a plurality of the blocks, specifically a multiple number (m) of the blocks. The sequence of blocks may be termed a partition, so that the length of the partition (k) equals the multiple times the block length (m)*(l). For example, the partition may comprise 4 blocks (m) and have a total length (k) of 4*32 bytes=128 bytes.

Jaquette et al., column 4, lines 53-61.

Jaquette does not calculate a running ECC check symbol. *Jaquette* does not calculate the ECC check symbol as the data is being written to the memory array. Therefore, *Jaquette* does not cure the deficiencies of *O'Grady*.

According to Applicant's claims, when writing to each one of the blocks is complete, a final value of the running cyclical redundancy code that was calculated for each block is stored as a first cyclical redundancy code. In *Jaquette*, an ECC check symbol is calculated after the data is formatted. *Jaquette* does not teach calculating a running cyclical redundancy code. *Jaquette* does not teach calculating a running cyclical redundancy code as data is being written. *Jaquette* does not teach storing the final value of a running cyclical redundancy code as a first cyclical redundancy code when writing the data is complete.

Indeed, the Examiner does not assert that *Jaquette* teaches "as said data is being written to each one of said plurality of blocks, calculating a running cyclical redundancy code for each of said plurality of blocks". On page 6 of the Final Office Action that was mailed February 7, 2007, the Examiner asserts that *Jaquette* teaches "calculating a first cyclical redundancy code (CRC) for each of said plurality of blocks (col. 4, ll. 24-27), wherein the first CRC corresponds to the ECC check symbols calculated".

Applicant does not claim a "first cyclical redundancy code" in this feature of the claim. Applicant claims a "running cyclical redundancy code". Therefore, because the Examiner admits that *O'Grady* does not teach this feature, and because the Examiner does not assert that *Jaquette* teaches this feature, the combination of *O'Grady* and *Jaquette* does not render Applicant's claims obvious.

Furthermore, teaching the calculation of an ECC for a block does not teach "as said data is being written to each one of said plurality of blocks, calculating a running cyclical redundancy code for each of said plurality of blocks". *Jaquette* waits until the data is formatted into blocks and then calculates a <u>final</u> ECC check symbol for the <u>entire</u> block of data. This ECC check symbol is not a <u>running</u> ECC check symbol that is calculated as data is being stored into the block. Therefore, the combination of *O'Grady* and *Jaquette* does not render Applicant's claims obvious.

The Examiner also does not assert that *Jaquette* teaches "when writing to each one of said plurality of blocks is completed, storing, for each one of said plurality of blocks, a final value of said running CRC that was calculated for each one of said plurality of blocks as a first CRC in a second memory on said buffering device". On page 6 of the Final Office Action that was mailed February 7, 2007, the Examiner asserts that *Jaquette* teaches "storing the first CRC in a second memory on said buffering device (col. 5, ll. 13-26), wherein the first CRC corresponds to the ECC check symbols are stored in a different segment of the DRAM from where the data blocks are stored".

Applicant does not claim "storing the first CRC in a second memory on said buffering device". Applicant claims "<u>when writing</u> to each one of said plurality of blocks <u>is completed</u>, storing, for each one of said plurality of blocks, <u>a final value of said running CRC</u> that was calculated for each one of said plurality of blocks <u>as a first CRC</u>". Therefore, because the Examiner admits that *O'Grady* does not teach this feature, and because the Examiner does not assert that *Jaquette* teaches this feature, the combination of *O'Grady* and *Jaquette* does not render Applicant's claims obvious.

Jaquette does not teach storing a final value of the running CRC for each block when writing to each block is completed. *Jaquette* teaches waiting until writing to a block is completed and then calculating and storing an ECC. This ECC is not a final value of another CRC. Therefore, the combination of *O'Grady* and *Jaquette* does not render Applicant's claims obvious.

On page 7 of the Final Office Action that was mailed February 7, 2007, the Examiner asserted that the combination of *O'Grady* and *Jaquette* teaches "as each cells of the data is being stored into the plurality of blocks, calculate the CRC for each of the cells being stored into each of the plurality of blocks, therefore calculating a running CRC". Applicant respectfully disagrees.

The Examiner states that *O'Grady* teaches allocating a plurality of blocks of storage by teaching a plurality of memory banks 160, 161. Each memory bank stores a single cell.

Since *Jaquette* teaches calculating an ECC for a block after data has been in the block, the combination of *O'Grady* and *Jaquette* would teach calculating an ECC for a memory bank after the cell had been written into the memory bank. An ECC would be calculated only after the cell had been completely written into the memory bank. Thus, a running ECC would not be calculated for the memory bank as the cell was being written into the memory bank.

Because the combination of *O'Grady* and *Jaquette* does not teach or suggest "as said data is being written to each one of said plurality of blocks, calculating a running cyclical redundancy code for each of said plurality of blocks; and when writing to each one of said plurality of blocks is completed, storing, for each one of said plurality of blocks, a final value of said running cyclical redundancy code that was calculated for each one of said plurality of blocks as a first cyclical redundancy code in a second memory on said buffering device", the combination does not render Applicant's claims obvious.

II.B. Claims 2 and 13 over *O'Grady* in view of *Jaquette* and further in view of *Hogan*

The Examiner has rejected claims 2 and 13 under 35 U.S.C. § 103(a) as being unpatentable over *O'Grady* in view of *Jaquette* and further in view of *Hogan* et al., Method and Apparatus for Discouraging Duplication of Digital Data, U.S. Patent No. 6,765,739, dated July 20, 2004 (hereinafter "*Hogan*"). This rejection is respectfully traversed.

Applicant's claims 2 and 13 recite similar features. Claim 2 recites "wherein the data is received with a protection code that is checked and discarded". The examiner relies on the combination of *Jaquette* and *Hogan* to cure the deficiencies of *O'Grady*.

The combination of *O'Grady*, *Jaquette*, and *Hogan* does not render claims 2 and 13 obvious because the combination does not teach as said data is being written to each one of said plurality of blocks, calculating a running cyclical redundancy code for each of said plurality of blocks; and when writing to each one of said plurality of blocks is completed, storing, for each one of said plurality of blocks, a final value of said running cyclical redundancy code that was calculated for each one of said plurality of blocks as a first cyclical redundancy code in a second memory on said buffering device in combination with the features of claims 2 and 13.. Therefore, the combination of *O'Grady*, *Jaquette*, and *Hogan* does not render Applicant's claims 2 and 13 obvious.

II.C. Claim 3 over *O'Grady* and *Jaquette* and further in view of *PCTechGuide*

The Examiner has rejected claim 3 under 35 U.S.C. § 103(a) as being unpatentable over *O'Grady* and *Jaquette* and further in view of *"PCTechGuide"*. This rejection is respectfully traversed.

Claim 3 recites wherein said buffering device is a DDR device connected between a bus and a tape drive. The combination of *O'Grady*, *Jaquette*, and *PCTechGuide* does not render Applicant's claim 3 obvious because the combination does not teach as said data is being written to each one of said plurality of blocks, calculating a running cyclical redundancy code for each of said plurality of blocks; and when writing to each one of said plurality of blocks is completed, storing, for each one of said plurality of blocks, a final value of said running cyclical redundancy code that was calculated for each one of said plurality of blocks as a first cyclical redundancy code in a second memory on said buffering device in combination with the features of claim 3. Therefore, the combination of *O'Grady*, *Jaquette*, and *PCTechGuide* does not render Applicant's claim 3 obvious.

II.D. Claims 7-8 and 10 over *O'Grady* in view of *Jaquette* and further in view of *Malakapalli*

The Examiner has rejected claims 7-8 and 10 under 35 U.S.C. § 103(a) as being unpatentable over *O'Grady* in view of *Jaquette* and further in view of *Malakapalli*, Mass Storage Error Correction and Detection System, Method and Article of Manufacture, U.S. Patent No. 6,467,060, dated October 15, 2002 (hereinafter "*Malakapalli*"). This rejection is respectfully traversed.

Applicant's independent claim 7 recites "said cyclical redundancy code engine calculating a running cyclical redundancy code for each one of said plurality of blocks as data is written to each one of said plurality of blocks" and "a second random access memory connected to said cyclical redundancy code engine for storing, for each one of said plurality of blocks, a final value of said running cyclical redundancy code that was calculated for each one of said plurality of blocks as a first cyclical redundancy code when writing to each one of said plurality of blocks is completed".

As discussed above, the combination of *O'Grady* and *Jaquette* does not teach calculating a running cyclical redundancy code for each block as data is written to each block; and storing, for each block, a final value of the running CRC that was calculated for each block when writing to each block is completed. *Malakapalli* does not cure the deficiencies of the combination of *O'Grady* and *Jaquette*. Therefore, the combination of *O'Grady*, *Jaquette*, and *Malakapalli* does not render Applicant's claim 7 obvious.

The remaining claims depend from claim 7 and are patentable for the reasons given above.

II.E. Claim 9 over *O'Grady*, *Jaquette* and *Malakapalli* and further in view of *Hogan*

The Examiner has rejected claim 9 under 35 U.S.C. § 103(a) as being unpatentable over *O'Grady*, *Jaquette* and *Malakapalli* in further view of *Hogan*. This rejection is respectfully traversed.

Claim 9 recites "a protection module connected to said first port for checking a protection code that is received and discarding said protection code". The combination of *O'Grady*, *Jaquette*, *Malakapalli*, and *Hogan* does not render Applicant's claim 9 obvious because the combination does not teach "said cyclical redundancy code engine calculating a running cyclical redundancy code for each one of said plurality of blocks as data is written to each one of said plurality of blocks" and "a second random access memory connected to said cyclical redundancy code engine for storing, for each one of said plurality of blocks, a final value of said running cyclical redundancy code that was calculated for each one of said plurality of blocks as a first cyclical redundancy code when writing to each one of said plurality of blocks is completed" in combination with the features of claim 9.. Therefore, the combination of *O'Grady*, *Jaquette*, *Malakapalli*, and *Hogan* does not render Applicant's claim 9 obvious.

III. <u>Conclusion</u>

It is respectfully urged that the subject application is patentable over the cited prior art and is now in condition for allowance.

The examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: April 9, 2007

Respectfully submitted,

Lisa L.B. Yociss
Reg. No. 36,975
Yee & Associates, P.C.
P.O. Box 802333
Dallas, TX 75380
(972) 385-8777
Attorney for Applicant